Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently amended) A multi-processor computer system, comprising:

~~a plurality of processors~~ a first processor and a second processor coupled together to permit messages to be transmitted from one processor to another processor;

~~each~~ said first processor having at least one timer that expires when a message is not sent from the first processor in a predetermined amount of time;

wherein ~~each~~ said first processor can send a plurality of different message types to ~~other of~~ said second processor[[s]] and ~~each such other~~ said second processor includes a separate timer associated with each of said message types to expire when a message of the associated message type is not sent in a predetermined amount of time from said second processor.

2.-4. (Canceled).

5. (Original) The multi-processor computer system of claim 1 further including at least one register associated with each timer to permit the timer to be programmed.

6. (Previously presented) The multi-processor computer system of claim 1 wherein each processor has at least one port connection to another processor and wherein each processor further includes a port timer associated with said port connection.

7. (Original) The multi-processor computer system of claim 6 wherein each port timer increments if the associated port is being used to send messages.

8. (Original) The multi-processor computer system of claim 7 wherein each port timer is reset when a message is sent from the port.

9. (Original) The multi-processor computer system of claim 7 wherein each port timer is reset when it receives a signal from a processor that receives a message from the port that indicates that the receiving processor has freed up an entry in an input buffer.

10. (Currently amended) A first processor that can be coupled to other processors to form a multi-processor system and can exchange messages with other processors in the system, ~~the~~ said first processor comprising:

router logic that can be coupled to at least one other processor;

said router logic having at least one timer that expires when a message is not sent from ~~the~~ said first processor in a predetermined amount of time; and

wherein ~~each~~ said first processor can send a plurality of different message types to other of said processors and each such other processor includes a separate timer associated with each of said message types to expire when a message of the associated message type is not sent in a predetermined amount of time .

11.-13. (Canceled).

14. (Original) The processor of claim 10 further including at least one register associated with each timer to permit the timer to be programmed.

15. (Previously presented) The processor of claim 10 wherein each processor has at least one port connection to another processor and wherein each processor further includes a port timer associated with said port connection.

16. (Original) The processor of claim 15 wherein each port timer increments if the associated port is being used to send messages.

17. (Original) The processor of claim 16 wherein each port timer is reset when a message is sent from the port.

18. (Original) The processor of claim 16 wherein each port timer is reset when it receives a signal from a processor that receives a message from the port that indicates that the receiving processor has freed up an entry in an input buffer.

19.-24. (Canceled).

25. (Currently amended) A method of averting message traffic congestion, comprising:

resetting a first timer in a processor when a message of a first type is sent, wherein the first timer expires if a message of the first type is not sent within a predetermined amount of time;

resetting a second timer in said processor when a message of a second type is sent wherein the second timer expires if a message of the second type is not sent within a predetermined amount of time; and

if the first or second timer expires, disabling transmission of messages of the corresponding type from said processor.

26. (Previously presented) The method of claim 25, further comprising:

incrementing the first timer while a buffer holds a message of the first type.

27. (Previously presented) The method of claim 26, further comprising:

incrementing the second timer while a buffer holds a message of the second type.

28. (Previously presented) The method of claim 25, further comprising:

incrementing a third timer while a buffer holds a message of a third type;

resetting the third timer when a message of the third type is sent; and

if the third timer expires, disabling transmission of messages of the third type to the buffer without simultaneously disabling transmission of messages of the first and second types.